Задача A. Детский сад

Имя входного файла:	`children.in`
Имя выходного файла:	`children.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

M детей в детском саду играют в следующую игру. Они рисуют на листочке бумаги N разноцветных кружочков (двух одинаковых не бывает, а то неинтересно), а затем каждый из них рисует по одной стрелочке из одного кружочка в другой. Когда кто-нибудь пытается нарисовать стрелочку, начинающуюся и кончающуюся в одном и том же кружочке, все остальные дружно возмущаются, мол, так нечестно. Также нельзя рисовать стрелочку, которая дублирует какую-нибудь уже нарисованную стрелочку (то есть, их начала и концы совпадают).

Когда каждый из них нарисует свою стрелочку, они все вместе собираются вокруг и смотрят, не образуют ли какие-нибудь кружочки хоровод. Несколько кружочков образуют хоровод, если они соединены стрелочками по кругу.

Вчера вечером дети тоже играли в эту игру. То ли потому что день был такой, то ли потому что поздно уже было, но каждый раз, что бы дети не делали, хоровод все время находился. Поэтому они все под конец расстроились и расплакались.

Под конец дня Вы пришли забрать брата из детского сада. Ваш брат тоже играл... Когда Вы спросили его, почему он плакал, он рассказал Вам, почему.

Вы в недоумении. Ведь существует столько способов нарисовать M стрелочек, чтобы дети не расстраивались!

Так сколько же?

Формат входного файла

Во входном файле в первой строке два числа — N и M. ($2 \le N \le 20,\ 0 \le M \le N^2 - N$)

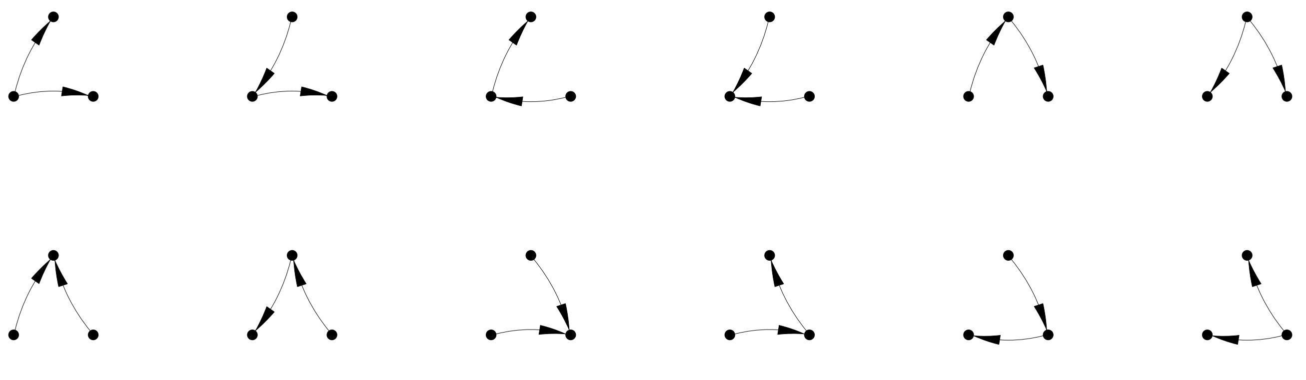
Формат выходного файла

В выходной файл выведите ответ на задачу.

Пример

children.in	children.out
`3 2`	`12`

Приведем все 12 возможных способов нарисовать стрелочки в примере из условия:



Задача B. Свободные от квадратов числа

Имя входного файла:	`sqfree.in`
Имя выходного файла:	`sqfree.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Число называется *свободным от квадратов*, если оно не делится ни на одно число вида x^2, где x — натуральное число, большее единицы.

Задано число n. Найдите наибольший его свободный от квадратов делитель (в число делителей включается единица и само число n).

Формат входного файла

Входной файл содержит целое число n ($1 \le n \le 10^{13}$).

Формат выходного файла

В выходной файл выведите ответ на задачу.

Примеры

sqfree.in	sqfree.out
9	3
20	10

Задача C. Следующая правильная скобочная последовательность

Имя входного файла:	`brackets.in`
Имя выходного файла:	`brackets.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Требуется найти следующую в лексикографическом порядке правильную скобочную последовательность.

Формат входного файла

Во входном файле задана правильная скобочная последовательность. Количество открывающихся скобок в последовательности — от 1 до 100 000.

Формат выходного файла

Выведите в выходной файл следующую в лексикографическом порядке среди всех правильных скобочных последовательностей с таким же количеством открывающихся скобок, «`(`» < «`)`». Если такой нет, выведите «`-`».

Пример

brackets.in	brackets.out
`(())()()`	`()((()))`

Задача D. Редакционное расстояние с блоками

Имя входного файла:	block.in
Имя выходного файла:	block.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	64 мегабайта

Редакционное расстояние или расстояние Левенштейна — универсальный инструмент, используемый, например, при сравнении файлов, или в вычислительной биологии и смежных науках для определения «похожести» строк. Однако у него есть недостаток, так как иногда оно не замечает длинные блоки, и символы, которые сопоставляются друг другу в строках часто слабо соответствуют смыслу изменений. Исправить этот недостаток помогает введение в редакционное расстояние понятия блока.

Вам заданы две строки S и T. Сначала вы можете выбрать некоторые непересекающиеся подстроки строки S (блоки) и удалить их. Удаление каждой подстроки стоит b. Пусть получилось слово Z.

После этого вы считаете обычное редакционное расстояние между Z и T. Редакционное расстояние вычисляется на базе последовательности инструкций, которые превращает Z в T. Разрешенные инструкции — "I" (вставить), "D" (удалить), и "C" (скопировать).

Рассмотрим два указателя, исходно первый указатель указывает на первый символ строки Z, а второй — на первый символ T. Инструкция "I" перемещает второй указатель на один символ вправо. Инструкция "D" перемещает первый указатель на один символ вправо. Инструкция "C" может применяться только если указатели указывают на один и тот же символ. Оба указателя при этом перемещаются направо. Если какой-либо указатель уже вышел за пределы своей строки, то использовать инструкцию, которая перемещает его направо, нельзя. Каждая инструкция "I" стоит i, каждая инструкция "D" стоит d, каждая инструкция "C" стоит c.

Найдите способ преобразовать S в T описанным способом за минимальную стоимость.

Формат входного файла

Первая строка входного файла содержит четыре целых числа: b, i, d и c ($0 \le b, i, d, c \le 10\,000$). Вторая строка содержит S. Третья строка содержит T. Длина каждой из S и T не превышает 3 000.

Формат выходного файла

Первая строка выходного файла должна содержать число k — стоимость превращения S в T описанным способом. Вторая строка должна содержать n — количество подстрок S, которые надо удалить. Следующие n строк должны содержать по два числа — индексы начала и конца подстрок (включительно), которые следует удалить. Символы нумеруются, начиная с 1.

Последняя строка должна содержать последовательность инструкций, чтобы превратить строку Z в T.

Пример

block.in	block.out
3 1 1 0 ABCDEFGHIJKLMN BCDEFZZZZKLM	9 1 7 10 DCCCCCIIIICCCD